identifying a routine whose input type matches the source type;

converting the data of the source type to the output type of the identified routine using the identified routine; and

until no more routines can be identified,

identifying a routine whose input type matches the output type resulting from the last conversion of the data; and

converting the data of the output type resulting from the last conversion of the data using the newly identified routine.

20. (New) The method of claim 19 wherein an input type matches an output type when an identifier of the input type and the output type are the same.

21. (New) The method of claims 19 wherein a type has an alias identifier and wherein the type matches another type when the alias identifier and an identifier of the other type are the same.

22. (New) The method of claims 19 wherein the identifying of a routine includes identifying multiple routines whose input type matches the output type for the last conversion of the data and wherein the converting of the data includes converting the data of the output type resulting from the last conversion of the data using each of the newly identified routines.

23. (New) The method of claim 22 wherein the identified routines form a tree-like arrangement of organization of routines.

24. (New) The method of claims 19 including terminating the identification of routines when the output type of an identified routine matches the target type.

25. (New) The method of claims 19 including caching an indication of the identified routines so that, when data of the source type to be converted to the target type is subsequently received, the data can be converted without re-identifying the routines.

26. (New) The method of claims 19 including storing in a cache an indication of a sequence of routines for converting data from the source type to the target type so that, when data of the source type to be converted to the target type is first received, the data can be converted using the sequence of routines indicated in the cache without identifying routines.

27. (New) The method of claims 19 including storing in a cache an indication of a sequence of routines for converting data from an input type so that, when data is converted from the input type, the data can be converted using the sequence of routines indicated in the cache without identifying the routines of the sequence.

28. (New) The method of claims 19 wherein when the target type is known, identifying a sequence of routines to convert the data from an input type to the target type prior to converting the data using any of the routines in the identified sequence.

29. (New) The method of claim 28 including, after identifying the sequence of routines, converting the data to the target type using the routines in the identified sequence.

30. (New) The method of claims 19 wherein a routine provides its output type only after converting data.

31. (New) The method of claims 19 wherein a routine sends its converted data of its output type to a forwarding component that identifies a routine and invokes the identified routine to convert the data.

32. (New) A computer system for converting data, comprising:
   a plurality of routines, each routine having an input type and an output type;
   a component that receives data to be converted, the data being of a source type; and
   a component that, after receiving the data, identifies a sequence of routines whose output types match the input types of the next routine in the sequence and converts the data of the source type using the identified sequence of routines.

33.　(New) The computer system of claim 32 wherein the data is converted using an identified routine before identifying the next routine in the sequence.

34.　(New) The computer system of claims 32 wherein an input type matches an output type when an identifier of the input type and the output type are the same.

35.　(New) The computer system of claims 32 wherein a type has an alias identifier and wherein the type matches another type when the alias identifier and an identifier of the other type are the same.

36.　(New) The computer system of claims 32 wherein the identifying of a routine includes identifying multiple routines whose input type matches the output type for the last conversion of the data and wherein the converting of the data includes converting the data of the output type resulting from the last conversion of the data using each of the newly identified routines.

37.　(New) The computer system of claim 36 wherein the identified routines form a tree-like arrangement of organization of routines.

38.　(New) The computer system of claims 32 wherein the identification of routines is terminated when the output type of an identified routine matches a target type of the received data.

39.　(New) The computer system of claims 32 including caching an indication of the identified routines so that, when data of the source type to be converted is subsequently received, the data can be converted without re-identifying the routines.

40.　(New) The computer system of claims 32 including storing in a cache an indication of a sequence of routines for converting data from the source type to a target type so that, when data of the source type to be converted to the target type is first received, the data can

be converted using the sequence of routines indicated in the cache without identifying routines in the sequence.

41.   (New) The computer system of claims 32 including storing in a cache an indication of a sequence of routines for converting data from an input type so that, when data is converted from the input type, the data can be converted using the sequence of routines indicated in the cache without identifying routines in the sequence.

42.   (New) The computer system of claims 32 wherein a routine provides its output type only after converting data.

43.   (New) The computer system of claims 32 wherein a routine sends its converted data of its output type to a forwarding component that identifies a routine and invokes the identified routine.

44.   (New) A computer-readable medium containing a data structure comprising a media class having a media class type and including includes a compare function for comparisons relating to the media class type.

45.   (New) The computer-readable medium of claim 44 wherein the compare function indicates whether is a media class type is equivalent to the media class type of the media class.

46.   (New) The computer-readable medium of claim 44 wherein the media class further includes an alias table indicating media class types equivalent to the class type of the media class.

47.   (New) The computer-readable medium of claim 46 wherein the alias type has an alias entry for each media type within the media class for which an alias has been registered.

48.   (New) The computer-readable medium of claim 47 wherein each alias entry contains a registered media for the media type.

49.   (New) The computer-readable medium of claim 44 wherein the media class further includes a protocol list indicating a protocol for processing data of the media class type of the media class.

50.   (New) A computer system for converting data, comprising:

a plurality of routines, each routine having an input type and an output type;

means for receiving data to be converted, the data being of a source type; and

means for, after receiving the data, identifying a sequence of routines whose output types match the input types of the next routine in the sequence and converts the data of the source type using the identified sequence of routines.

51.   (New) The computer system of claim 50 including a media class for each type.

52.   (New) The computer system of claim 51 wherein each media class includes a compare function.

53.   (New) The computer system of claim 52 wherein the compare function identifies matching types.

54.   (New) The computer system of claim 50 wherein the data is converted using an identified routine before identifying the next routine in the sequence.

55.   (New) The computer system of claims 50 wherein an input type matches an output type when an identifier of the input type and the output type are the same.

56.   (New) The computer system of claims 50 wherein a type has an alias identifier and wherein the type matches another type when the alias identifier and an identifier of the other type are the same.

57.   (New) The computer system of claims 50 wherein the identifying of a routine includes identifying multiple routines whose input type matches the output type for the last